AD-A119 234 BROWN UNIV PROVIDENCE RI LEFSCHETZ CENTER FOR DYNAMI--ETC F/G 12/1
PARAMETER ESTIMATION IN TIMOSHENKO BEAM MODELS.(U)
JUN 82 H T BANKS, J M CROWLEY DAAG29-79-C-0161
UNCLASSIFIED LCDS-82-14 AFOSR-TR-82-0721 NL

END
DATE
FILMED
10-82
DTIC

Lefschetz Center for Dynamical Systems

Division of Applied Mathematics

Brown University Providence RI 02912

②

PARAMETER ESTIMATION
IN
TIMOSHENKO BEAM MODELS

by

H.T. Banks and J.M. Crowley

June 1982 LCDS Report #82-14

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 82-0721	2. GOVT ACCESSION NO. AD-A117234	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PARAMETER ESTIMATION IN TIMOSHENKO BEAM MODELS		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL
		6. PERFORMING ORG. REPORT NUMBER LCDS #82-14
7. AUTHOR(s) H.T. Banks and J.M. Crowley		8. CONTRACT OR GRANT NUMBER(s) AFOSR-81-0198
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lefschetz Center for Dynamical Systems, Division of Applied Mathematics, Brown University, Providence RI 02912		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A4
11. CONTROLLING OFFICE NAME AND ADDRESS Directorate of Mathematical & Information Sciences Air Force Office of Scientific Research Bolling AFB DC 20332		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 30
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The authors present cubic and linear spline-based approximation schemes for models of beams based on the Timoshenko theory. The schemes are used in parameter estimation algorithms; convergence results and numerical findings are reported.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PARAMETER ESTIMATION IN TIMOSHENKO BEAM MODELS*

by

H.T. Banks and J.M. Crowley+
Lefschetz Center for Dynamical Systems
Division of Applied Mathematics
Brown University
Providence, RI 02912

June 1982

*Work supported in part by the National Science Foundation under grant NSF-MCS 79-05774-05, in part by the U.S. Army Research Office under grant contract ARO-DAAG-29-79-C-0161 and in part by the Air Force Office of Scientific Research under contract AFOSR 81-0198.

+Part of this research was carried out while both authors were visitors at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA which is operated under NASA contracts No. NAS1-15810 and No. NAS1-16394.

PARAMETER ESTIMATION IN
TIMOSHENKO BEAM MODELS

H.T. Banks and J.M. Crowley

ABSTRACT

We present cubic and linear spline-based approximation schemes for models of beams based on the Timoshenko theory. The schemes are used in parameter estimation algorithms; convergence results and numerical findings are reported.

PARAMETER ESTIMATION IN TIMOSHENKO BEAM MODELS

Introduction

When modeling large complex structures, it is often desirable to model them as a simple structure such as a beam or plate. For example, a large latticed structure with many components may be efficiently modeled as a single beam using either the Euler-Bernoulli or Timoshenko theory (see [1], [2], [3]), depending on the importance of shear effects and rotary inertia. The identification of structural parameters (e.g. bending and shear rigidity, rotatory inertia) plays an important role when modeling large complex structures in this manner. Identification or estimation is important in model development and verification and can also play a fundamental role in procedures for design of feedback controllers.

The purpose of this paper is to propose numerical methods for estimation of constant parameters in the Timoshenko equations for transverse vibrations of a beam, to report convergence results for these methods and to give some numerical results demonstrating the effectiveness of the associated estimation algorithms. We restrict our attention to the convergence properties of the methods and do not address such important questions as identifiability, observability, or global stability of the solutions of the underlying partial differential equations. For the presentation here, we employ observations consisting of discrete measurements of transverse displacements, although our theory is sufficiently general to include problems with measurements of velocity or strain in place of displacements.

The theory and methods presented here have also been successfully applied to estimation of parameters in the Euler-Bernoulli equation [4], [5] and in other hyperbolic and parabolic partial differential equations [6] - [10]. The equation to which we restrict our attention here, however, is the Timoshenko system [11, p. 300]:

$$(1) \qquad \begin{aligned} y_{tt} &= a(y_{xx} - \psi_x) + f(t,x;\tilde{q}) \\ \psi_{tt} &= b(y_x - \psi) + c\psi_{xx} \end{aligned} \quad , \quad 0 \le x \le 1, \quad t > 0,$$

where $y(t,x)$ is the transverse displacement and $\psi(t,x)$ is the angle of rotation of a cross-section of the beam. Here $a = k'AG/m$, $c = EA/m$, $b = Aa/I$ with A = cross sectional area, E = Young's modulus, G = shear modulus, I = moment of inertia, k' = shear coefficient, m = mass per unit length, and f represents the transverse loading (possibly depending on unknown parameters $\tilde{q}$).

Various boundary conditions are of interest, depending on the physical situation. Among these are:

(2)
(a) $y(t,0) = \psi(t,0) = y(t,1) = \psi(t,1) = 0$ (fixed end beam)
(b) $y(t,0) = \psi_x(t,0) = y(t,1) = \psi_x(t,1) = 0$ (simply supported beam)
(c) $y(t,1) = \psi(t,1) = y_x(t,0) - \psi(t,0) = \psi_x(t,0) = 0$ (centilevered beam).

In addition one must specify initial conditions $y(0,x)$, $y_t(0,x)$, $\psi(0,x)$, $\psi_t(0,x)$ for (1).

Suppose that we are given a set of observations $\{\hat{y}_{ij}\}$ that correspond to measurements of the transverse displacements $\{y(t_i,x_j)\}$ at times t_i and at points x_j. We assume that the structure that we are observing can be modeled by equation (1). We seek those values of certain unknown

parameters q (either unknown coefficients in (1) and unknown parameters in the load function and/or in the initial conditions) so that the solution $(y,\psi) = (y(q),\psi(q))$ of (1) best fits the data $\{\hat{y}_{ij}\}$ in some sense. Here we shall measure the fit of the model to the data by a sum of squares functional of the form:

$$J(q,y) = \sum_{i,j} |\hat{y}_{ij} - y(t_i,x_j)|^2.$$

The parameter identification problem is then to find from a given parameter set Q, values $\bar{q}$ of the parameters which minimize $J(q,y)$ subject to y being a solution of (1).

Since explicit solutions of (1) are, in general, impossible to find, we are led to seek approximate solutions of the parameter estimation problem. We first approximate solutions of (1) by some appropriate method to obtain approximate solutions (y^N,ψ^N). The associated estimation problem then consists of finding a vector of parameters $\bar{q}^N$ which minimizes $J(q,y^N)$. Our goal here is to present a class of methods for estimation of parameters for which we have proven convergence of the solutions of the approximating problems. That is, we have shown that $\bar{q}^N \longrightarrow \bar{q}$ as $N \to \infty$. Further, we hope to demonstrate numerical feasibility of these methods.

The class of methods we examine for approximating solutions to (1) are projection or Ritz-Galerkin type methods. The initial-boundary value problem for (1) is rewritten in terms of a system of first order operator differential equations in an appropriate Hilbert space Z. This abstract system

$$(3) \qquad \begin{aligned} \dot{z}(t) &= A(q)z(t) + F(q,t) \\ z(0) &= z_0 \end{aligned}$$

is then approximated by a system of the form

$$\text{(4)} \qquad \begin{aligned} \dot{z}^N(t) &= A^N(q)z^N(t) + F^N(q,t) \\ z^N(0) &= P^N z_0, \end{aligned}$$

with $A^N(q) \equiv P^N A(q) P^N$ and $F^N = P^N F$, where P^N is the orthogonal projector of the Hilbert space Z onto some finite dimensional subspace Z^N. We shall, in each case discussed below, choose Z^N as the linear span of a particular set of spline functions satisfying the appropriate boundary conditions. For these schemes, we have established convergence as $N \to \infty$ of the parameter estimates to their best-fit values (for details, see [4], [5], [6]).

Formulation of the Abstract System

The Timoshenko equations (1) can be rewritten as a first order system of partial differential equations in a number of ways. Certain reformulations are more suitable to handling given boundary conditions than others. Each reformulation leads to a different abstract system (that is, the operator A in (3) and perhaps the Hilbert space Z depend on the particular reformulation) and thus can lead to a different approximating system. We examine two such reformulations here. The first we investigate is most natural when considering beams with fixed ends (i.e. boundary conditions (2a)). We formally rewrite (1) with $z = (v_1, v_2, v_3, v_4) = (y, y_t, \psi, \psi_t)$ as

$$
(5)\qquad
\begin{aligned}
\frac{\partial}{\partial t} v_1 &= v_2 \\
\frac{\partial}{\partial t} v_2 &= a \frac{\partial^2}{\partial x^2} v_1 - a \frac{\partial}{\partial x} v_3 + f(t,x;\tilde{q}) \\
\frac{\partial}{\partial t} v_3 &= v_4 \\
\frac{\partial}{\partial t} v_4 &= b \frac{\partial}{\partial x} v_1 + c \frac{\partial^2}{\partial x^2} v_3 - b v_3 \ .
\end{aligned}
$$

Letting $z_i(t) = v_i(t,\cdot)$, this formulation leads to the abstract equation (3) in $Z = H_0^1 \times L^2 \times H_0^1 \times L^2$, with $q = (a,b,c,\tilde{q}) = (q_1,q_2,q_3,\tilde{q})$, where

$$
(6)\qquad A(q) = \begin{pmatrix} 0 & I & 0 & 0 \\ q_1 D^2 & 0 & -q_1 D & 0 \\ 0 & 0 & 0 & I \\ q_2 D & 0 & q_3 D^2 - q_2 I & 0 \end{pmatrix}, \quad D = \frac{\partial}{\partial x},
$$

on $\mathrm{Dom}(A(q)) = (H^2 \cap H_0^1) \times L^2 \times (H^2 \cap H_0^1) \times L^2$, and

$$
F(q,t) = \begin{pmatrix} 0 \\ f(t,\cdot;\tilde{q}) \\ 0 \\ 0 \end{pmatrix} .
$$

Cantilever boundary conditions cannot be handled easily in this formulation; another reformulation is more appropriate. Using the change of variables $v_1 = y_x - \psi$, $v_2 = y_t$, $v_3 = \psi_x$, $v_4 = \psi_t$, $v_5 = y$, one can obtain a system of first order partial differential equations

$$\frac{\partial v_1}{\partial t} = \frac{\partial v_2}{\partial x} - v_4$$

$$\frac{\partial v_2}{\partial t} = a\,\frac{\partial v_1}{\partial x} + f(t,x;\tilde{q})$$

$$(7) \qquad \frac{\partial v_3}{\partial t} = \frac{\partial v_4}{\partial x}$$

$$\frac{\partial v_4}{\partial t} = bv_1 + c\,\frac{\partial v_3}{\partial x}$$

$$\frac{\partial v_5}{\partial t} = v_2,$$

which, under the correspondence $z_i(t) = v_i(t,\cdot)$, leads to the abstract equation (3) in $Z = L^2 \times L^2 \times L^2 \times L^2 \times L^2$, with $q = (a,b,c,\tilde{q})$ where

$$F(q,t) = \begin{pmatrix} 0 \\ f(t,\cdot;\tilde{q}) \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and

$$(8) \qquad A(q) = \begin{pmatrix} 0 & D & 0 & -I & 0 \\ q_1 D & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D & 0 \\ q_2 I & 0 & q_3 D & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{pmatrix}$$

on $\mathrm{Dom}(A(q)) = H_L^1 \times H_R^1 \times H_L^1 \times H_R^1 \times H_R^1$. Here the spaces $H_L^1 = \{\phi \in H^1\colon \phi(0) = 0\}$ and $H_R^1 = \{\phi \in H^1\colon \phi(1) = 0\}$ are defined to deal with the appropriate boundary conditions $v_1(t,0) = v_3(t,0) = v_2(t,1) = v_4(t,1) = v_5(t,1) = 0$.

One may view the techniques employed to approximate the partial differential equations above as a particular Galerkin approximation to (5) or

(7). Our formulation as an abstract equation in a Hilbert space Z provides a convenient setting in which to prove convergence of the systems.

Convergence Results

As we have already stated, once we have the initial-boundary value problem for (1) rewritten as an abstract system (3) in an appropriate Hilbert space Z, we approximate (3) using projections $P^N: Z \longrightarrow Z^N$. In particular we choose a sequence of finite dimensional subspaces Z^N, with $Z^N \subset \mathrm{Dom}(A)$, such that $P^N z \to z$ for all $z \in Z$, and use the sequence of approximating ordinary differential systems (4).

Given this approximation, we can formulate an approximate identification or estimation problem: find $\bar{q}^N$ which minimizes $J(q,y^N)$ over Q, where $y^N(t,\cdot) = z_1^N(t)$ with z^N the solution of (4).

We outline briefly (for further details on the theory, see [4], [5], [6], [12]) the essential ideas involved in establishing convergence as $N \to \infty$ of the estimates $\bar{q}^N$ to best-fit parameter values $\bar{q}$ for (3). In our approach, the concept of dissipativeness plays a central role. An operator A is said to be dissipative ([13], [14], [15]) if there exists a constant ω such that $\langle Az,z\rangle \leq \omega\langle z,z\rangle$ for every z in $\mathrm{Dom}(A)$. It is this property of the operator $A(q)$ (and not self-adjointness) which we require when formulating the abstract system and making convergence arguments.

To prove convergence one first shows that $A(q)$ satisfies a uniform (in q) dissipative inequality in Z and that $A(q)$ generates a C_0 semigroup $T(t;q)$. In the cases of the operators in (6) and (8), the latter is accomplished using standard results on perturbation of semigroups. The

approximating operators $A^N(q)$, which are defined by $A^N(q) = P^N A(q) P^N$, are easily shown to be uniformly (in N and q) dissipative and to generate a stable family of approximating solution semigroups $e^{A^N t}$ such that $|\exp[A^N(q)t]| \leq Me^{\omega t}$, where M and ω are independent of N and q for q in a bounded set. Standard estimates from spline interpolation theory may be used to obtain bounds on the error of the projections $|P^N z - z|$. The projection error bounds then in turn are used to show that $A^N(q)z \longrightarrow A(q)z$ in an appropriate sense. One then employs the Trotter-Kato theorem [13, p. 90] (a functional analytic version of the Lax Equivalence Theorem; stability + consistency ➡ convergence) to argue that $[\exp A^N(q)t]\, z \longrightarrow T(t;q)z$ and, moreover, that $z^N(t;q) \longrightarrow z(t;q)$ where z^N and z are solutions of (4) and (3) respectively. These convergence results can then be used to argue that any sequence $\bar{q}^N$ of solutions to the approximating estimation problems (involving (4)) has a subsequence $\bar{q}^{N_k}$ that converges to some $\bar{q}$ that is a solution of the original estimation problem for (3).

Implementation and Numerical Results

As noted above, to implement our methods one begins by writing the underlying initial-boundary value problem as an abstract system in an appropriately chosen (Hilbert) state space. To do this, one writes the partial differential equation (1) in the form of a system of first order partial differential equations (such as (5) or (7)) and chooses the appropriate state space in which to pose the problem as an abstract initial value problem. Of course, one must check that the chosen formulation leads to a well-posed problem and that the resulting operator $A(q)$ generates a

C_0 semigroup $T(t;q)$ in Z (we have done this for all cases discussed in this paper).

Given this formulation in a space Z, one then chooses the approximating subspaces Z^N in $\mathrm{Dom}(A(q))$. A standard Galerkin approach may be used to obtain a concrete realization of the approximating system (4). To illustrate, we first consider the equation (5) in $Z = H_0^1 \times L^2 \times H_0^1 \times L^2$. We use as basis for our approximating space Z^N a set of cubic splines which satisfy the boundary conditions.

To be more precise, given a partition $\Delta^N = \{x_i\}_{i=0}^N$, $x_i = i/N$, of $[0,1]$, a cubic spline is any function $s \in C^2[0,1]$ such that s is a cubic polynomial on each subinterval (x_i,x_{i+1}), $i = 0,\ldots,N-1$. We denote by $S^3(\Delta^N)$ the set of cubic splines with knots Δ^N. A cubic spline interpolant to a function $f \in C^1$ on $[0,1]$ is the function $s \in S^3(\Delta^N)$ satisfying $Ds(\eta) = Df(\eta)$ for $\eta = 0$ and 1, and $s(x_i) = f(x_i)$, $i = 0,1,\ldots,N$. As a basis for $S^3(\Delta^N)$, the standard B-splines [16, p. 89] can be employed: Define

$$\overline{C}(x) \equiv \sum_{i=-2}^{2} (-1)^{i+2} \binom{4}{i+2} (x-i)_+^3, \qquad x \in [-2,2]$$

where

$$(x-i)_+^3 = \begin{cases} (x-i)^3 & x \geq i \\ 0 & x < i. \end{cases}$$

Then define

$$\hat{C}_i^N(x) = \overline{C}((x-x_i)/h) \tag{9}$$

where $h = 1/N$. Defining an extended partition $\Delta_*^N = \{x_i\}_{i=-1}^{N+1}$, the set of cubic B-splines $\{\hat{C}_i^N\}_{i=-1}^{N+1}$ provide a basis for $S^3(\Delta^N)$. In anticipation of our consideration of boundary conditions of type (2a) in our ensuing dis-

cussions, we define $S_0^3(\Delta^N) = \{s \in S^3(\Delta^N) | s(0) = s(1) = 0\}$. As a basis for $S_0^3(\Delta^N)$ we take $\{C_i^N\}_{i=0}^N$ where

$$C_i^N = \hat{C}_i^N, \qquad 2 \leq i \leq N-2,$$

$$C_0^N = \hat{C}_0^N - 4\hat{C}_{-1}^N \qquad C_N^N = \hat{C}_N^N - 4\hat{C}_{N+1}^N$$

$$C_1^N = \hat{C}_0^N - 4\hat{C}_1^N \qquad C_{N-1}^N = \hat{C}_N^N - 4\hat{C}_{N-1}^N.$$

For the approximating subspaces Z^N we choose $Z^N = S_0^3(\Delta^N) \times S_0^3(\Delta^N) \times S_0^3(\Delta^N) \times S_0^3(\Delta^N)$. With C_i^N as the basis elements just defined for $S_0^3(\Delta^N)$, we obtain as a basis for Z^N the set $\{\beta_i^N\}_{i=0}^{4N+3}$ where

$$\beta_i^N = \begin{cases} (C_i^N,0,0,0), & i = 0,\ldots,N \\ (0,C_{i-(N+1)}^N,0,0), & i = N+1,\ldots,2N+1 \\ (0,0,C_{i-(2N+2)}^N,0), & i = 2N+2,\ldots,3N+2 \\ (0,0,0,C_{i-(3N+3)}^N), & i = 3N+3,\ldots,4N+3. \end{cases}$$

Given Z^N, in the Galerkin approach to our method one seeks $z^N \in Z^N$ satisfying

$$\langle \dot{z}^N - A^N z^N - P^N F, \beta \rangle = 0 \quad \text{for all} \quad \beta \in Z^N, \tag{10}$$

where $\langle , \rangle$ is the inner product in Z. It suffices to require (10) for $\beta = \beta_j^N$, $j = 0, \ldots, 4N+3$. Since $z^N \in Z^N$ it can be represented by $z^N = \sum w_i^N(t)\beta_i^N$ and the condition (10) thus reduces to

$$\langle \sum \dot{w}_i^N \beta_i^N - \sum w_i^N A\beta_i^N - F, \beta_j^N \rangle = 0 \tag{11}$$

for $j = 0,1,\ldots,4N+3$. This Ritz-Galerkin argument leads to a $4N+4$-dimensional matrix system for the coefficients $w_i^N(t)$ in the expansion for $z^N(t)$ relative to the basis for Z^N. In particular, one finds that $w^N(t) = (w_0^N(t),\ldots,w_{4N+3}^N(t))$ satisfies

$$Q^N \dot{w}^N(t) = K^N w^N(t) + R^N F$$

with $(Q^N)_{ij} = \langle \beta_i, \beta_j \rangle$, $(K^N)_{ij} = \langle \beta_i, A\beta_j \rangle$, $(R^N F)_i = \langle \beta_i, F \rangle$. This becomes

(12) $$\dot{w}^N(t) = G^N w^N(t) + F^N$$

with

$$G^N = \begin{bmatrix} 0 & I & 0 & 0 \\ q_1 (A_2^N)^{-1} A_1^N & 0 & -q_1 (A_2^N)^{-1} A_3^N & 0 \\ 0 & 0 & 0 & I \\ q_2 (A_2^N)^{-1} A_3^N & 0 & q_3 (A_2^N)^{-1} A_1^N - q_2 I & 0 \end{bmatrix}$$

where $(A_2^N)_{i,j} = \langle C_i^N, C_j^N \rangle$, $(A_1^N)_{i,j} = -\langle DC_i^N, DC_j^N \rangle$, $(A_3^N)_{i,j} = \langle C_i^N, DC_j^N \rangle$, (here $\langle\ ,\ \rangle$ is the L_2 inner product),

$$(F^N)_i = \begin{cases} f_i^N, & i = N+1, \dots, 2N+1 \\ 0, & \text{otherwise} \end{cases}$$

where $f^N = (f_0^N, \dots, f_N^N)$ corresponds to the "load" f in (5) and is given by $f^N = (A_2^N)^{-1} (\hat{R}^N f)$ with $(\hat{R}^N f)_i = \langle C_i^N, f \rangle$. Finally, since $z_1(t) \equiv y(t,\cdot)$, the approximation to the displacement y is given by

(13) $$y^N(t,\cdot) = z_1^N(t) = \sum_{i=0}^{N} w_i(t) C_i^N.$$

We next summarize results of some of our numerical experiments using the approximation scheme based upon cubic splines just described. But first we describe how our numerical tests were performed. We took as "data" the values of the solution of a model of the form (1) whose parameters $\bar{q}$ were known (i.e. chosen by us) and sought a solution $\bar{q}^N$ of the approximate identification problem for different values of N. Specifically, the "data"

$\{\hat{y}_{ij}\}$, $i = 1,\ldots,r$, $j = 1,\ldots,\ell$ were generated in this case using the L^2 formulation of the Timoshenko equations (5) and using a general purpose computer code (MOL1D) to solve this system of first-order hyperbolic equations to obtain displacements $y(t,x)$. Then $\{\hat{y}_{ij}\}$, $i = 1,\ldots,r$, $j = 1,\ldots,\ell$ with $\hat{y}_{ij} = y(t_i,x_j)$ were used as the observations or input to the approximate identification package. For the numerical experiments detailed below, we used $\ell = 9$, $r = 10$ and generated the "observations" at $x_j = j/10$, $j = 1,\ldots,9$ for times $t_i = i/10$, $i = 1,\ldots,10$.

For each fixed N, an optimization algorithm (the Levenberg-Marguardt in the IMSL package ZXSSQ) was employed to solve the approximate estimation problem. This optimization algorithm requires an initial guess $q^{N,0}$ for the parameters, which is referred to as the "start up" value in the tables below. The values of the parameters to which the optimization algorithm converged for a given N are denoted by $\bar{q}^N$ and $J(\bar{q}^N)$ is the cost functional (residual sum of squares) for those values of the parameters, while the values $\bar{q}$ are called "true values".

<u>Example 1</u>. We consider the motion of a beam initially at rest with fixed ends. That is, our beam is described by the system (1) with $a = q_1$, $b = q_2$, $c = q_3$, $f = 10e^{-2t}\sin 2t$, boundary condition (2a) and vanishing initial data. The numerical results are given in Table 1.

Table 1

N	$\bar{q}_1^N$	$\bar{q}_2^N$	$\bar{q}_3^N$	$J(\bar{q}^N)$
6	.9882	726.19	3.6864	$.526 \times 10^{-4}$
8	.9969	781.00	3.9218	$.735 \times 10^{-5}$
10	1.0009	794.29	3.9684	$.108 \times 10^{-5}$
12	1.00036	794.90	3.9732	$.165 \times 10^{-6}$
16	1.00033	797.85	3.9883	$.256 \times 10^{-7}$
TRUE VALUE	1.0	800.	4.0	
START UP	.9	1000.	3.9	

Example 2. We consider a clamped beam deformed to the shape $\phi(x) = \cos \lambda x + \cosh \lambda x - K(\sin \lambda x + \sinh \lambda x)$ with $\lambda \simeq 4.730$, $K = (\sin \lambda + \sinh \lambda)/(\cos \lambda + \cosh \lambda)$, then allowed to vibrate freely, which can be described by the system consisting of equation (1) with $f = 0$, initial conditions $y(0,x) = \phi(x)$, $y_t(0,x) = 0$, $\psi(0,x) = \phi'(x)$, $\psi_t(0,x) = 0$, and boundary conditions (2a). The numerical results are given in Table 2. As in Example 1, the methods perform exceedingly well.

Table 2

N	$\bar{q}_1^N$	$\bar{q}_2^N$	$\bar{q}_3^N$	$J(\bar{q}^N)$
4	1.1812	1325.3	3.6437	$.613 \times 10^{-2}$
8	.9480	1125.1	3.993	$.167 \times 10^{-2}$
16	.9908	1152.3	3.8875	$.242 \times 10^{-3}$
24	1.0009	1193.7	3.9771	$.122 \times 10^{-3}$
TRUE VALUE	1.0	1200.	4.0	
START UP	.9	800.	3.8	

We next consider examples for the formulation (7), which, as we have already indicated, is very convenient when treating cantilevered beams. Recall (see (8)) in this case $\mathrm{Dom}(A(q)) = H_L^1 \times H_R^1 \times H_L^1 \times H_R^1 \times H_R^1$ and that the cantilever boundary conditions are $v_1(t,0) = v_3(t,0) = v_2(t,1) = v_4(t,1) = v_5(t,1) = 0$. For this formulation one can easily use either cubic or linear splines. We first outline the procedure using cubic elements.

The set of cubic splines $S^3(\Delta^N)$ and its basis of $N+3$ B-splines $\{\hat{C}_i^N\}_{i=-1}^{N+1}$ were defined in (9) above. These were then modified to yield basis elements satisfying particular boundary conditions. We do the same again here. To obtain a basis for Z^N whose elements satisfy the cantilever boundary conditions (observe that the projection method requires that they be in $\mathrm{Dom}(A)$), we define

$$S_L^3(\Delta^N) = \{s \in S^3(\Delta^N): \ s(0) = 0\}$$
$$S_R^3(\Delta^N) = \{s \in S^3(\Delta^N): \ s(1) = 0\}$$

and as a basis for $S_L^3(\Delta^N)$ take $\{C_i^N\}_{i=0}^{N+1}$ where $C_i^N = \hat{C}_i^N$, $2 \le i \le N+1$,

$$C_0^N = \hat{C}_0^N - 4\hat{C}_{-1}^N$$
$$C_1^N = \hat{C}_0^N - 4\hat{C}_1^N$$

(the $\hat{C}_i^N$ are the B-splines given in (9) above).

As a basis for $S_R^3(\Delta^N)$ take $\{B_i^N\}_{i=0}^{N+1}$ where

$$B_i^N = \hat{C}_{i-1}^N, \quad i = 0,\dots,N-1,$$
$$B_{N+1}^N = \hat{C}_N^N - 4\hat{C}_{N+1}^N$$
$$B_N^N = \hat{C}_N^N - 4\hat{C}_{N-1}^N.$$

Now we take $Z^N = S_L^3 \times S_R^3 \times S_L^3 \times S_R^3 \times S_R^3$ in the case of cubic spline approximations.

Turning to approximation schemes employing linear splines, we use usual "hat" functions

$$\ell_i^N(x) = \begin{cases} N(x - x_{i-1}), & x_{i-1} \leq x \leq x_i \\ N(x_{i+1} - x), & x_i \leq x \leq x_{i+1} \\ 0 \quad , & \text{otherwise} \end{cases}$$

where $\Delta^N = \{x_i\}_{i=0}^N$ are the knots as defined in setting up the cubic splines. Denote by $S^1(\Delta^N)$ the set of linear splines with knots Δ^N; i.e., $s \in S^1(\Delta^N)$ is a function $s \in C[0,1]$ such that s is a linear function on each subinterval $[x_i,x_{i+1}]$, $i = 0,\ldots,N-1$. The $N+1$ functions $\{\ell_i^N\}_{i=0}^N$ form a basis for $S^1(\Delta^N)$. Define $S_L^1(\Delta^N) = \text{span}\{\ell_1^N,\ldots,\ell_N^N\}$ and $S_R^1(\Delta^N) = \text{span}\{\ell_0^N,\ldots,\ell_{N-1}^N\}$. Then S_L^1 and S_R^1 are N-dimensional subspaces of linear splines in H_L^1 and H_R^1 respectively. We take $Z^N = S_L^1 \times S_R^1 \times S_L^1 \times S_R^1 \times S_R^1$ for the linear spline approximations.

The usual Ritz-Galerkin formulation leads to a 5N+10-dimensional matrix system of ordinary differential equations for the coefficients $w_i^N(t)$ in the expansion for $z^N(t)$ relative to the cubic spline basis for Z^N and a 5N-dimensional system in the linear spline case. For the cubic spline case, the approximating matrix equations (a system of $5N + 10$ ordinary differential equations in this case) again have the form

$$Q^N \dot{w}^N(t) = K^N w^N(t) + F^N(q,t)$$

where now Q^N and K^N are $5N + 10$ square matrices defined by

$$Q^N = \begin{pmatrix} Q_1^N & & & & \\ & Q_2^N & & 0 & \\ & & Q_1^N & & \\ & 0 & & Q_2^N & \\ & & & & Q_2^N \end{pmatrix},$$

with

$$(Q_1^N)_{ij} = \langle C_i^N, C_j^N \rangle$$

$$(Q_2^N)_{ij} = \langle B_i^N, B_j^N \rangle, \quad i,j = 0,\ldots,N+1,$$

and

$$K^N = \begin{pmatrix} 0 & K_4^N & 0 & -K_3^N & 0 \\ -q_1(K_4^N)^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_4^N & 0 \\ q_2(K_3^N)^T & 0 & -q_3(K_4^N)^T & 0 & 0 \\ 0 & Q_2^N & 0 & 0 & 0 \end{pmatrix},$$

with

$$(K_3^N)_{ij} = \langle C_i^N, B_j^N \rangle$$

$$(K_4^N)_{ij} = \langle C_i^N, DB_j^N \rangle, \quad i,j = 0,\ldots,N+1,$$

(again, here $\langle\ ,\ \rangle$ is the inner product in L_2), and

$$F^N = (0, f^N, 0, 0, 0)^T,$$

$$(f^N)_i = \langle f(t,\cdot;\tilde{q}), B_i^N \rangle,$$

and $f(t,x;\tilde{q})$ is the "load" term in (7). The matrices Q_i^N are 7-banded and symmetric while the matrices K_i^N are 7-banded but not symmetric.

For the linear spline case, the approximating system of 5N ordinary differential equations has the same structure as above with C_i^N replaced by ℓ_{i+1}^N, $i = 0,\ldots,N-1$, and B_i^N by ℓ_i^N, $i = 0,\ldots,N-1$. The $5N \times 5N$ matrices Q_i^N and K_i^N however are only tridiagonal in the linear case.

Example 3. In our first example with cantilever boundary conditions presented here, we used values for the coefficients reported in [1] where a cantilever truss system is "modelled" as a continuous beam. That is, we choose as true values $a = \bar{q}_1 = 298.257$, $b = \bar{q}_2 = 413520.$, $c = \bar{q}_3 = 3590.57$ in the equations (1) with $f = \exp(-2t-2x)$, initial conditions for a beam at rest, and with cantilever boundary conditions (2c). Again, simulated data (with observations at $x_j = j/10$, $t_i = i/10$, $j = 0,\ldots,9$, $i = 1,\ldots,10$) were generated by solving (7) using the general purpose PDE package MOL1D.

Of course, in practice one might be interested in using fewer sensors or points x_j where data is sampled. Recall that our original intent is to examine the convergence properties of a numerical parameter estimation scheme and we wish to avoid questions of observability, identifiability, and optimal location of sensors. As a comparison, however, we have included below another example where data is sampled at one point only (the tip of the cantilever). The following three tables summarize the results for this example using both the cubic and linear spline schemes. In each of these experiments, only one parameter was treated as unknown and the other two were fixed at their true value.

Table 3.1

(treating q_1 as the unknown parameter)

N	Linear Spline $\bar{q}_1^N$	Cubic Spline $\bar{q}_1^N$
2	339.913	297.792
3	283.762	298.032
4	310.843	298.109
5	292.163	
6	300.690	
8	302.266	
10	299.072	
TRUE VALUE	298.257	298.257
START UP	200.	200.

Table 3.2

(treating q_2 as the unknown parameter)

N	Linear Spline $\bar{q}_2^N$	Cubic Spline $\bar{q}_2^N$
1		415363.
2	351677.	414597.
3		413632.
4	391564.	413584.
5	398172.	
6	405907.	
8	409217.	
10	410404.	
TRUE VALUE	413520.	413520.
START UP	360000.	360000.

Table 3.3

(treating q_3 as the unknown parameter)

$\underline{N}$	Linear Spline $\bar{q}_3^N$	Cubic Spline $\bar{q}_3^N$
1		3553.57
2	4261.79	3590.95
3	3410.55	3587.77
4	3779.85	3590.50
6	3684.83	
10	3541.13	
TRUE VALUE	3590.57	3590.57
START UP	3000.00	3000.00

We note that both the linear and cubic spline approximations converge rapidly and give reliable estimates even for small values of N.

The optimization portion of the computer codes (IMSL's ZXSSQ) encountered difficulties in this example when we attempted to estimate all three parameters simultaneously, due to the large difference in magnitude of the parameters. This can be corrected by a rescaling in the optimization routine.

Example 4. We consider next another example involving cantilever boundary conditions. For this example we compare not only cubic versus linear spline approximations, but also compare results with a large number of sensors (ten points, $\{x_i\}_{i=0}^{9}$, $x_i = i/10$) to those obtained in the case of a small number of sensors (one point, $x_0 = 0$).

The Timoshenko equations (1) with the beam initially at rest and with cantilever boundary conditions (2c) was the model. The applied load

f had the form $f(t,x) = \exp(-5t)\exp(-4x)\sin(2t)$. Simulated data was again generated employing the general purpose PDE solver MOL1D using the "true values" $a = \bar{q}_1 = 1.0$, $b = \bar{q}_2 = 1.2 \times 10^3$, $c = \bar{q}_3 = 12.0$. We note that parameter values on this scale are obtained when a second change of variables (in addition to rescaling the spatial variable to the interval [0,1] as in [1]) is performed by rescaling the time variable $t = \alpha t'$ where t' is the original time scale, and α^2 is of the same order of magnitude as q_1. Two sets of "observation" data were employed. Both used the value of the approximate solution $\hat{y}_{ij} = y(t_i,x_j)$ at $t_i = i/10$, $i = 1,\ldots,10$; the number of spatial points used were:

(A) ten points $x_i = i/10$, $i = 0,\ldots,9$;

(B) one point $x_0 = 0$ (at the free tip).

The following tables summarize the results obtained. Table 4.1 through 4.4 involve use of data of the form (A), while Tables 4.5 through 4.8 result from use of data of type (B). Thus Tables 4.5 - 4.8 present results using only observations at the tip (free end) of the cantilever beam at ten times ($t_i = i/10$, $i = 1,\ldots,10$).

Table 4.1

(treating q_1 as the unknown parameter)

N	Linear Spline $\bar{q}_1^N$	Cubic Spline $\bar{q}_1^N$
1		1.05917
2		1.01066
3	1.52290	.99549
4	1.18569	1.00094
5	1.15071	
6	1.05861	
8	1.02103	
12	1.00702	
TRUE VALUE	1.0	1.0
START UP	.8	.8

Table 4.2

(treating q_2 as the unknown parameter)

N	Linear Spline $\bar{q}_2^N$ (values × 10^3)	Cubic Spline $\bar{q}_2^N$ (values × 10^3)
1		1.10840
2		1.19366
3	.76109	1.20853
4	.98753	1.19892
5	1.02709	
6	1.12289	1.20043
8	1.17090	
12	1.18983	
16	1.19399	
TRUE VALUE	1.2×10^3	1.2×10^3
START UP	$.8 \times 10^3$	$.8 \times 10^3$

Table 4.3

(treating q_3 as the unknown parameter)

N	Linear Spline $\bar{q}_3^N$	Cubic Spline $\bar{q}_3^N$
1		12.9822
2		12.0636
3	19.1429	11.9116
4	14.5998	12.0110
5	14.0550	12.0118
6	12.8214	11.9956
8	12.2942	
12	12.0991	
16	12.0581	
TRUE VALUE	12.0	12.0
START UP	10.0	10.0

Table 4.4

(treating all three parameters as unknown)

Linear Spline

N	$\vec{q}_1^{\,N}$	$\vec{q}_2^{\,N}$ ($\times 10^3$)	$\vec{q}_3^{\,N}$	$J(\vec{q}^{\,N})$
3	1.68674	1.61611	14.4218	$.197\times10^{-2}$
4	3.38199	1.68874	5.4154	$.166\times10^{-2}$
6	1.06937	.65044	6.3526	$.478\times10^{-3}$
8	.99744	.78523	7.9986	$.152\times10^{-3}$
12	1.00496	.98184	9.2938	$.328\times10^{-4}$
16	1.00972	1.03337	10.2396	.99[illegible]$\times10^{-5}$
TRUE VALUE	1.0	1.2×10^3	12.0	
START UP	.8	$.8\times10^3$	10.0	

Cubic Spline

N	$\vec{q}_1^{\,N}$	$\vec{q}_2^{\,N}$ ($\times 10^3$)	$\vec{q}_3^{\,N}$	$J(\vec{q}^{\,N})$
2	2.29673	1.65908	6.6929	$.561\times10^{-4}$
3	1.27137	1.16502	8.8100	$.161\times10^{-4}$
4	1.01927	1.19287	11.6788	$.507\times10^{-5}$
6	1.01317	1.20155	11.9933	$.957\times10^{-7}$
8	1.00067	1.20160	12.0051	$.470\times10^{-8}$
TRUE VALUE	1.0	1.2×10^3	12.0	
START UP	.8	$.8\times10^3$	10.0	

Table 4.5

(q_1 treated as unknown)

N	Linear Spline $\vec{q}_1^{\,N}$	Linear Spline $J(\vec{q}^{\,N})$	Cubic Spline $\vec{q}_1^{\,N}$	Cubic Spline $J(\vec{q}^{\,N})$
1			1.08757	$.563\times10^{-3}$
2			.97849	$.194\times10^{-3}$
3	1.43945	$.950\times10^{-3}$	.97627	$.377\times10^{-4}$
4	1.27313	$.151\times10^{-2}$	1.00126	$.967\times10^{-5}$
5	1.16280	$.744\times10^{-3}$		
6	1.09265	$.647\times10^{-3}$		
8	1.04641	$.256\times10^{-3}$		
TRUE VALUE	1.0		1.0	
START UP	.8		.8	

Table 4.6

(q_2 treated as unknown)

	Linear Spline		Cubic Spline	
N	$\bar{q}_2^N(\times 10^3)$	$J(\bar{q}^N)$	$\bar{q}_2^N(\times 10^3)$	$J(\bar{q}^N)$
1			1.07420	$.502\times 10^{-3}$
2			1.23856	$.189\times 10^{-3}$
3	.810693	$.108\times 10^{-3}$	1.23247	$.370\times 10^{-4}$
4	.919741	$.158\times 10^{-2}$	1.19859	$.967\times 10^{-5}$
5	1.01736	$.747\times 10^{-3}$	1.19745	$.105\times 10^{-5}$
6	1.08323	$.636\times 10^{-3}$	1.20103	$.131\times 10^{-6}$
8	1.14092	$.255\times 10^{-3}$		
10	1.16499	$.111\times 10^{-3}$		
12	1.17608	$.551\times 10^{-4}$		
16	1.18724	$.180\times 10^{-4}$		
TRUE VALUE	$1.2\text{x}10^3$		$1.2\text{x}10^3$	
START UP	$1.0\text{x}10^3$		$1.0\text{x}10^3$	

Table 4.7

(q_3 treated as unknown)

	Linear Spline		Cubic Spline	
N	$\bar{q}_3^N$	$J(\bar{q}^N)$	$\bar{q}_3^N$	$J(\bar{q}^N)$
1			13.3809	$.512\times 10^{-3}$
2			11.5821	
3	17.8588	$.118\times 10^{-2}$	11.6669	$.365\times 10^{-4}$
4	15.6401	$.165\times 10^{-2}$	12.0138	$.968\times 10^{-5}$
5	14.1666	$.774\times 10^{-3}$		
6	13.2724	$.651\times 10^{-3}$	11.9894	$.131\times 10^{-6}$
8	12.6035	$.259\times 10^{-3}$	11.9954	$.557\times 10^{-8}$
10	12.3605	$.112\times 10^{-3}$		
12	12.2367	$.558\times 10^{-4}$		
16	12.1228	$.194\times 10^{-4}$		
TRUE VALUE	12.0		12.0	
START UP	8.0		8.0	

Table 4.8

(treating all three parameters as unknown)

Linear Spline

N	$\bar{q}_1^N$	$\bar{q}_2^N(\times 10^3)$	$\bar{q}_3^N$	$J(\bar{q}^N)$
3	2.25943	.313872	1.59287	$.181\times 10^{-3}$
4	4.16515	.37462	.83466	$.105\times 10^{-3}$
6	.95858	.12019	1.09886	$.565\times 10^{-4}$
8	.74126	.13653	1.67134	$.283\times 10^{-4}$
12	1.062306	.360775	3.17335	$.697\times 10^{-5}$
16	1.03356	.504339	4.68815	$.209\times 10^{-5}$
20	1.04769	.652659	6.08683	$.108\times 10^{-5}$
24	1.02567	.745678	7.13885	$.442\times 10^{-6}$
28	1.01353	.820307	7.98051	$.241\times 10^{-6}$
TRUE VALUE	1.0	1.2×10^3	12.0	
START UP	.8	1.0×10^3	8.0	

Cubic Spline

N	$\bar{q}_1^N$	$\bar{q}_2^N(\times 10^3)$	$\bar{q}_3^N$	$J(\bar{q}^N)$
2	2.62212	1.69692	5.97524	$.879\times 10^{-4}$
3	2.31452	1.80666	7.23897	$.212\times 10^{-4}$
4	1.00842	1.20156	11.91970	$.963\times 10^{-5}$
TRUE VALUE	1.0	1.2×10^3	12.0	
START UP	.8	1.0×10^3	8.0	

The machine time and computations required in use of our algorithms are difficult to give precisely because this depends on many factors: starting values (initial guesses for the q_i), convergence criteria for the optimization method, and error tolerances in the ODE solver. Since our primary purpose has been to examine convergence properties, the convergence criteria employed were very stringent (for example, convergence to 5 decimal digit accuracy in $\bar{q}^N$) and thus our computations were not

made with the greatest regard for efficiency. However, it is clear that even our less-than-optimal programs can be fine-tuned to give more efficient solutions when less accuracy is needed.

To give some measure of the amount of computation required and as a basis of comparison between linear and cubic spline schemes as used in the examples above, we note the typical times required (on the IBM 370 at Brown University) for each evaluation of $J(q,y^N)$ in the above examples ranged from 1.5 CPU seconds ($N = 3$) to 11 seconds ($N = 16$) for the linear splines while ranges for the cubic scheme were 2.5 seconds ($N = 1$) to 7.5 seconds ($N = 6$).

To compare the relative efficiency of the linear versus cubic splines and to provide a further measure of the time involved, we list the following results for Example 4, with data of type (A). As we have noted, the Levenberg-Marquardt algorithm was used to perform the optimization for a given approximation level N. It is an iterative procedure, and each iteration may require many evaluations of J to evaluate the gradient and to test if J has been decreased after each new iterative step. We report the number of iterative steps, evaluations of J, seconds of CPU time per evaluation of J and total CPU time to find $\bar{q}^N$ from the starting value reported in Tables 4.1 - 4.3. These results are pessimistic in the sense that all convergence parameters were set to obtain maximal accuracy, resulting in more iterative steps. However, it does serve as something of a basis for comparison. When treating q_1 as unknown, the following CPU time was required (see Table 4.1).

approx	N	iterative steps	evaluations of J(q)	CPU seconds per evaluation	total CPU seconds
Linear	3	5	13	1.8	25.7
	4	5	13	2.2	31.4
	5	3	23	2.7	64.4
	6	3	9	3.2	29.7
	8	4	15	5.0	76.5
Cubic	1	5	15	2.5	37.5
	2	3	9	3.0	27.7
	3	3	9	4.0	36.0

When treating q_2 as unknown, the following CPU time was required (see Table 4.2).

approx	N	iterative steps	evaluations of J(q)	CPU seconds per evaluation	total CPU seconds
Linear	3	7	30	1.5	49.6
	4	4	16	2.2	37.0
	6	3	9	3.6	34.4
	8	3	9	4.4	44.3
	12	3	9	7.2	66.2
Cubic	1	3	9	2.5	22.9
	2	3	9	3.0	27.7
	3	3	9	5.0	47.3

When treating q_3 as unknown, the following CPU time was required (see Table 4.3).

approx	N	iterative steps	evaluations of J(q)	CPU seconds per evaluation	total CPU seconds
Linear	3	5	13	1.7	23.2
	4	4	16	2.4	39.8
	5	2	15	2.7	42.0
	6	3	9	3.5	31.9
	8	3	9	5.0	46.4
	10	3	9	6.0	57.5
	12	3	9	7.0	65.4
Cubic	2	5	9	3.0	27.0
	3	3	9	4.0	36.3
	4	3	9	5.0	45.3

In most cases, the cubic spline approximation was more efficient than linear splines. For example, when treating q_2 as unknown, the cubic splines with $N = 2$ required 27.7 seconds of CPU time to find $\bar{q}_2^N$; to achieve the same accuracy with linear splines, it was necessary to use $N = 12$ which required 66.2 seconds of CPU time. In the case when all three parameters were treated as unknown, the relatively greater efficiency of the cubic splines is even more important. The linear splines failed to give a good approximation even for $N = 16$. Some typical times to solve this problem when all 3 parameters were treated as unknown are (Table 4.4):

approx	N	iterative steps	evaluations of J(q)	time per evaluation	total time
Linear	8	9	31	4.0	126.
	16	7	27	11.0	297.
Cubic	8	5	21	12.0	258.

In summary, both the cubic spline and linear spline approximation schemes converged rapidly when many spatial observations (i.e. data of type (A)) were available and both gave good estimates even for small values of N. For high accuracy estimates, the cubic spline method was more efficient. For problems with limited data (type (B) of Example 4), the difference in performance of the methods is more marked. The linear spline estimates for all three parameters simultaneously appeared to be converging, but failed to give good estimates of q_2 and q_3 even for $N = 28$; the cubic approximation on the other hand gave a very accurate estimate for $N = 4$ in far less time.

Thus the linear approximations are satisfactory when high accuracy is not required and when data is sufficient to permit linear approximations. Otherwise the cubic spline approximations provide an excellent alternative which we believe is computationally feasible.

Conclusions

In this paper we have presented parameter estimation methods that should prove useful in modeling large complex structures as continua. We have focused on systems described by the Timoshenko theory although models for shear beams (see [1]) or beams with pure bending (the Euler-Bernoulli theory) can also be easily treated with our techniques (see [4]). Our methods have a sound theoretical foundation (i.e. we can prove convergence) and we have successfully tested them numerically on a large number of examples (some of our findings being reported above).

We are currently developing the methods (both theoretical investigations and computational tests offer encouragement) for elastic models with spatially varying parameters (coefficients). We are optimistic that these methods will prove extremely useful in engineering studies of large structures that are currently of interest in aeronautics and astronautics.

References

[1] Sun, C. T., Kim, B. J., and Bogdanoff, J. A., On the derivation of equivalent simple models for beam-and plate-like structures in dynamic analysis, Proc. AIAA Dynamics Specialists Conf., Atlanta, April 6-8, 1981, pp. 523-532.

[2] Chen, C. C., and Sun, C. T., Transient analysis of large frame structures by simple models, J. Astronautical Sci., to appear.

[3] Juang, J. N., and Sun, C. T., System identification of large flexible structures by using simple continuum models, to appear.

[4] Banks, H. T., and Crowley, J. M., Parameter estimation for distributed systems arising in elasticity, LCDS Tech. Rep. 81-24, Brown Univ., 1981, in Proc. NCKU/AAS Symposium on Engineering Sciences and Mechanics, Tainan, Taiwan, Dec. 28, 1981 - Jan. 1, 1982.

[5] Crowley, J.M., Numerical Methods of Parameter Identification for Problems Arising in Elasticity, Ph.D. Thesis, Brown University, May, 1982.

[6] Banks, H. T., Crowley, J. M., and Kunisch, K., Cubic spline approximation techniques for parameter estimation in distributed systems, LCDS Tech. Rep. 81-25, Brown Univ., 1981, IEEE Trans. Auto Control, to appear.

[7] Banks, H. T., A survey of some problems and recent results for parameter estimation and optimal control in delay and distributed parameter systems, LCDS Tech. Rep. 81-19, Brown Univ., 1981, Proc. Conf. on Volterra and Functional Differential Equations, Blacksburg, Va., June 10-13, 1981.

[8] Banks, H. T., Distributed system optimal control and parameter estimation: computational techniques using spline approximations, LCDS Tech. Rep. 82-6, Brown Univ., 1982, Proc. 3rd IFAC Symposium on Control of Distributed Parameter Systems, Toulouse, France, June 29-July 2, 1982.

[9] Sives, C., and Sato, L., Computer estimation of parameters in a brain fluid transport equation: A cubic spline approximation method, LCDS Tech. Rep. M-82-6, Brown Univ., 1982.

[10] Banks, H. T., and Kareiva, P., Parameter estimation techniques for transport equations with applications to population dispersal and tissue bulk flow models, LCDS Tech. Rep. 82-13, Brown Univ., 1982.

[11] Clough, R. W. and Penzien, J., Dynamics of Structures, McGraw-Hill, New York, 1975.

[12] Banks, H. T., and Kunisch, K., An approximation theory for nonlinear partial differential equations with applications to identification and control, LCDS Tech. Rep. 81-7, Brown Univ., 1981; SIAM J. Control and Optimization, to appear.

[13] Pazy, A., Semigroups of Linear Operators and Applications to Partial Differential Equations, Lec. Notes No. 10, Univ. Maryland, 1974.

[14] Balakrishnan, A. V., Applied Functional Analysis, Springer, New York, 1976.

[15] Krein, S. G., Linear Differential Equations in Banach Space, Transl. Math. Momo. Vol. 29, Amer. Math. Soc., Providence, R.I., 1971.

[16] Prenter, P. M., Splines and Variational Methods, J. Wiley, New York, 1975.